SMS: Smart Model Selection in PhyML
Vincent Lefort, Jean-Emmanuel Longueville, Olivier Gascuel

To cite this version:

Vincent Lefort, Jean-Emmanuel Longueville, Olivier Gascuel. SMS: Smart Model Selection in PhyML. Molecular Biology and Evolution, Oxford University Press (OUP), 2017, 34 (9), pp.2422-2424. 10.1093/molbev/msx149. lirmm-01794206

HAL Id: lirmm-01794206
https://hal-lirmm.ccsd.cnrs.fr/lirmm-01794206

Submitted on 13 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License
SMS: Smart Model Selection in PhyML

Vincent Lefort, Jean-Emmanuel Longueville, and Olivier Gascuel*\textsuperscript{1,2}
\textsuperscript{1}Institut de Biologie Computationnelle, LIRMM, UMR 5506 - CNRS et Université de Montpellier, Montpellier, France
\textsuperscript{2}Unité de Bioinformatique Evolutive, C3BI, USR 3756 - Institut Pasteur et CNRS, Paris, France

*Corresponding author: E-mail: olivier.gascuel@pasteur.fr.
Associate editor: Tal Pupko

Abstract

Model selection using likelihood-based criteria (e.g., AIC) is one of the first steps in phylogenetic analysis. One must select both a substitution matrix and a model for rates across sites. A simple method is to test all combinations and select the best one. We describe heuristics to avoid these extensive calculations. Runtime is divided by \(\sim 2\) with results remaining nearly the same, and the method performs well compared with ProtTest and jModelTest\textsuperscript{2}. Our software, “Smart Model Selection” (SMS), is implemented in the PhyML environment and available using two interfaces: command-line (to be integrated in pipelines) and a web server (http://www.atgc-montpellier.fr/phyml-sms/).

Key words: model selection, heuristic procedure, AIC and BIC criteria, web server, PhyML.
Computations with DNA are simpler than with proteins, as today’s MSAs are most often large enough for GTR to be best compared to other substitution matrices. Moreover, the simplest matrices are not satisfactory because they do not account for the transition/transversion ratio and/or unequal base frequencies. Experiments with 500 representative MSAs confirmed these hypotheses, and are congruent with the large-scale study of (Arbiza et al. 2011). With AIC, GTR is best for 343/500 MSAs, whereas JC69, K80, and F81 are all best with 9/500 MSAs only (supplementary table S3, Supplementary Material online). With BIC, K80 is best for 48/500 MSAs. SMS thus uses four substitution matrices: GTR, TN93, HKY85, and K80, which are combined with +I, +Γ, +Γ +I, and “none” (all four RAS options are useful, supplementary table S3, Supplementary Material online), that is, a total of 4 x 4 = 16 models. On average, SMS computes the likelihood value of ~6 models with AIC and 7.5 with BIC, thus dividing the computing time by ~2 as compared to the exhaustive approach using the same models. Based on the user’s selected criterion (AIC/BIC), the basic principle in SMS as follows: i) using a BioNJ tree topology, SMS estimates the branch lengths and model parameters for GTR and the four RAS options; ii) using the “most promising” RAS option with GTR, SMS selects the best matrix in a stepwise manner: SMS compares GTR and TN93; if GTR is better, then SMS stops and keeps GTR; otherwise, SMS compares HKY85 to TN93, and so on (remember that GTR, TN93, HKY85, and K80 are nested); iii) SMS selects the best RAS option for the best matrix. This simple approach, combined with a relatively small set of models, makes SMS nearly as fast as jModelTest2.
average AIC/BIC difference is in favor of SMS. With DNA, a better model than SMS in cases; when the models differ (35/500 MSAs), ProtTest finds proteins, SMS and ProtTest find the same model in most able in ProtTest and jModelTest2 (with default options). With MtZoa for proteins and TN93 for DNA, which are not avail-

table in ProtTest, partly explained by its set of substitution matrices, notably to tailored heuristics. The gains in AIC/BIC with SMS are better models than jModelTest2 with the fast “model filtering” option (supplementary fig. S1, Supplementary Material online).

Despite substantial gains in computing time, the results of SMS are nearly the same as those obtained with the exhaustive approach using the same models, and SMS performs well compared with jModelTest2 and ProtTest (table 1). To benchmark these methods, we used 500 DNA and 500 protein MSAs, corresponding to the first MSAs submitted to the PhyML Web server since the beta test version of SMS was made available (April 2015). No selection was performed, so these data sets are representative of the MSAs commonly used for phylogenetic analyses. Some of these MSAs are very small (e.g., 231 amino acids in total, with 11 taxa, and 231 sites); some are very large (e.g., 14,160,098 amino acids; some contain more than 1,000 taxa; and some have a huge number of sites (e.g., 52,092 nucleotidic sites). To confirm our findings, we also reused the 100 medium-size MSAs used to benchmark PhyML 3.0 (Guindon et al. 2010). The results with this second, independent set of MSAs, are fully congruent (supplementary table S6, Supplementary Material online).

We launched jModelTest2 and ProtTest with fast options, since SMS was designed to be fast. Moreover, we selected the options to make these two programs as close as possible to SMS in terms of substitution matrices, RAS modeling, and equilibrium frequency estimation. The results are shown in table 1. To summarize: SMS performs well compared with the exhaustive approach, in most cases finding identical or similar models regarding AIC/BIC values, whereas the gain in computing time is quite substantial. Moreover, SMS tends to select better models than jModelTest2 with the fast “model filtering” option, and is much faster than ProtTest, thanks to tailored heuristics. The gains in AIC/BIC with SMS are partly explained by its set of substitution matrices, notably MtZoa for proteins and TN93 for DNA, which are not available in ProtTest and jModelTest2 (with default options). With proteins, SMS and ProtTest find the same model in most cases; when the models differ (35/500 MSAs), ProtTest finds a good model, whereas the average AIC/BIC difference is in favor of SMS. With DNA, the sets of models are more different than with proteins, and SMS and jModelTest2 differ for 120 and 192 MSAs with AIC and BIC, respectively, whereas the gain in computing time gains of SMS with proteins are quite substantial in practice (supplementary fig. S1, Supplementary Material online). For example, ProtTest requires more than 100 h to process the largest MSA (1,151 taxa and 798 sites), whereas SMS requires ∼20 h using the same computer.

### Supplementary Material

Supplementary data are available at Molecular Biology and Evolution online.

### Acknowledgment

This research was supported by the Institut Français de Bioinformatique (RENABI-IFB, Investissements d’Avenir).

### References


### Table 1. Method Comparison with 500 DNA, and 500 Protein Representative MSAs.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Data</th>
<th>Criterion</th>
<th>Same Model</th>
<th>SMS Better</th>
<th>SMS Worse</th>
<th>Δ AIC &amp; Δ BIC per taxon per site</th>
<th># PhyML Runs</th>
<th>Speed Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS versus Exhaustive</td>
<td>DNA</td>
<td>AIC</td>
<td>486</td>
<td>na</td>
<td>14</td>
<td>4.6 x 10⁻⁵</td>
<td>6.1/16</td>
<td>1.9–2.0</td>
</tr>
<tr>
<td></td>
<td>BIC</td>
<td></td>
<td>476</td>
<td>na</td>
<td>24</td>
<td>8.0 x 10⁻⁵</td>
<td>7.5/16</td>
<td>1.7–1.9</td>
</tr>
<tr>
<td>SMS versus Exhaustive</td>
<td>Protein</td>
<td>BIC</td>
<td>494</td>
<td>na</td>
<td>6</td>
<td>3.7 x 10⁻³</td>
<td>29.3/68</td>
<td>2.2–2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIC</td>
<td>497</td>
<td>na</td>
<td>3</td>
<td>3.8 x 10⁻³</td>
<td>30.2/68</td>
<td>2.1–2.0</td>
</tr>
<tr>
<td>SMS versus jModelTest2</td>
<td>DNA</td>
<td>AIC</td>
<td>380</td>
<td>85</td>
<td>35</td>
<td>-2.5 x 10⁻⁵</td>
<td>6.1/17.8</td>
<td>1.1–0.8</td>
</tr>
<tr>
<td></td>
<td>BIC</td>
<td></td>
<td>308</td>
<td>151</td>
<td>41</td>
<td>-1.1 x 10⁻⁴</td>
<td>7.5/7.8</td>
<td>0.9–0.8</td>
</tr>
<tr>
<td>SMS versus ProtTest</td>
<td>Protein</td>
<td>AIC</td>
<td>465</td>
<td>14</td>
<td>21</td>
<td>-8.9 x 10⁻⁴</td>
<td>29.3/120</td>
<td>3.7–3.4</td>
</tr>
<tr>
<td></td>
<td>BIC</td>
<td></td>
<td>465</td>
<td>12</td>
<td>23</td>
<td>-7.5 x 10⁻⁴</td>
<td>30.2/120</td>
<td>3.5–3.2</td>
</tr>
</tbody>
</table>

Note.—The “Exhaustive” approach uses the same set of models as SMS and evaluates all of them. “Same model”: number of times (among 500 MSAs) where both methods return the same model; “SMS better”: number of times where the model returned by SMS has a lower AIC/BIC value; “SMS worse”: number of times where the model returned by SMS has a higher AIC/BIC value; “Δ AIC and Δ BIC per taxon per site”: when both models were different, we computed the difference in AIC/BIC per taxon per site, and averaged the results over all MSAs showing a model difference (a negative/positive value means that SMS’s model is better/worse in terms of AIC/BIC); “# PhyML runs”: number of PhyML runs for one method versus the other; “Speed increase”: for each MSA, we computed the computing time ratio of the method being compared with respect to SMS (e.g., 2 means that SMS is twice as fast), with the column displaying i) the median value among the 500 speedup ratios for all MSAs, ii) the median value for the 50 largest MSAs (number of sites x number of taxa; see supplementary fig. S1, Supplementary Material online for additional computing time results with large MSAs).